Assignment 2: 2D Animation

Submission due: Monday, October 13, 2025, 11:59PM

Please note that many sections are the same as in the previous assignment. For your convenience, the sections that have been changed are marked with an asterisk (*).

1. Introduction*

In this assignment, you will extend the previous assignment by applying hierarchical animation.
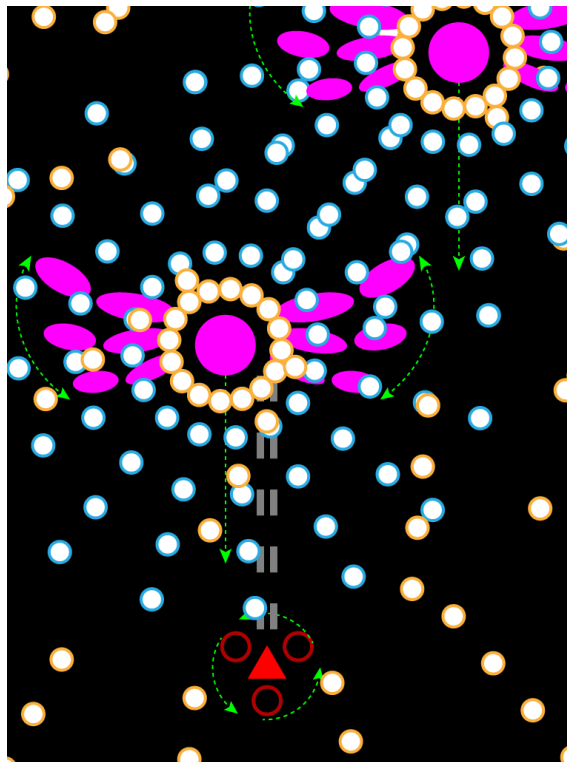

Figure 1. Bullet Hell shooter enhanced with hierarchical animation.

For inspiration, You may refer to examples of Bullet Hell shooters with hierarchical animation:

- Hijiri Byakuren's wings in Touhou 12th Undefined Fantastic Object
- R'lyeh's gun barrels in Hitogata Happa
- "Deep One"'s arms in Hitogata Happa
- Muffet boss animating in Undertale

- [Asriel Dreemurr boss animating in Undertale](#)

2. Requirements*

Minimum Goals*

Assignment 2 must satisfy all the minimum requirements of Assignment 1. In case of any conflicts, the requirements specified in Assignment 2 take precedence.

- Player
  - The player has orbiting entities surrounding them.
  - The number of entities reflects the player's remaining lives.
- Enemy
  - There are two or more enemies in this assignment, and they may be of the same kind.
  - The enemies appear at the top of the screen and move slowly toward the bottom.
  - The enemies' appearance must be represented using hierarchical animation.
    - The appearance does not need to match the enemies' functionality. For example, if enemies have moving arms, you do not need to make bullets emit from the tips of the arms; they may instead be emitted from the center.

Additional Goals*

To receive full credit, you must establish at least two additional goals beyond the minimum requirements above. These goals should be sufficiently challenging. Examples include:

- Multi-level (more than 2) hierarchical animation
- Parametric shaping of entities, e.g., the shape of enemies changes as time passes or as their location changes.

Note that credit will be given **only for graphics-related features**. Little or no credit will be given for mere game features (e.g., pause menus, gamepad support, multi-stage boss fights, etc.). You may choose additional goals from the examples above, but remember that more generous credit will be given for original goals you propose yourself.

Non-goals*

- You are not required to render fancy sprites in this assignment. It is enough to render entities as simple polygons.

3. Submission

Each team must submit one report along with the program's source code.

Report

Your report can be any document file format. However, please do **not** use `*.hwp`. Your report should be brief and concise, and include the following:

- Basic information:
  - The name of the team.
  - Each team member's name, department, student ID, and HEMOS ID.
- Technical details:
  - The development environment used (e.g., IDEs, framework versions, etc.).
  - Any additional technical background necessary to understand the source code.
- Implementation details:
  - An outline of your program's features.
  - Detailed description of your additional requirements.
  - The rationale behind your program's design.
  - How you implemented your design.
  - Any additional background necessary to understand the design and implementation.
  - Do **not** include your program's source code in the report.
- End-user guide:
  - How to run and operate your program.
  - Screenshots of each feature you described.
- Discussions/Conclusions:
  - The obstacles you encountered during development and how you resolved them.
  - Idea on how you can improve your program further.
  - What you learned or concluded from this assignment.
- References:
  - **Clearly state references** for any work not created by your team. (e.g., something from tech blogs, Stack Overflow posts, etc.)

- **If any part of your assignment is found to be someone else's work without proper references, it will be considered cheating.**
- **AI-assisted coding references**:
  - If you used any AI-assisted coding tools in writing your program (e.g., OpenAI Codex, Claude Code, Google Gemini, Cursor, Amazon Kiro, GitHub Copilot, etc.), clearly state which tools you used, how you used them, and where in your program they were applied.
  - You should also mention AI tools you consulted, even if their outputs were ultimately not included in your final program.
  - Provide an estimate of what percentage of your assignment was completed with AI assistance, and explain your reasoning. *This percentage will not affect your credit at all*.

Program

- Include a brief `README.md` file that describes your source code files.
- Your source code is expected to meet the technical requirements specified in the provided setup document.

4. Scoring Criteria

Basic Rules

- **Minimum requirements** (40%)
  - The program must run properly and meet the minimum requirements.
- **Additional requirements** (20% + *10% extra credit*)
  - You must define and implement additional requirements on your own.
  - Two valid additional requirements → +20% credit.
  - Three or more valid additional requirements → up to +30% total (20% base + 10% extra).
  - You may propose more than the minimum if you are unsure whether certain requirements will qualify for full credit.
- **Program design and implementation** (30%)
  - The program must be designed and implemented properly to meet the requirements using OpenGL.
- **Report structure and `README.md`** (10%)
  - The report and `README.md` must be clear and concise.
  - The structure and formatting of the report will also be evaluated.

Deduction

- If you miss the submission deadline, your score will be reduced by 10%.
- For every additional 24 hours late, your score will be reduced by another 10%p.
- For example,
    - Submitting 8 hours late: -10%
    - Submitting 25 hours late: -20%
    - Submitting 216 hours (9 days) late: no credit will be given.
- Note: After the submission deadline, you may only submit via the TA's email (`yoonha.hwang@postech.ac.kr`).

If you have any further questions, please post your question on Q&A board on PLMS.